

```

UUU          UUU  EEEEEEEEEEEEEEEEE  TTTT TTTT TTTT TTTT  PPPPPPPPPPPPP  PPP
UUU          UUU  EEEEEEEEEEEEEEEEE  TTTT TTTT TTTT TTTT  PPPPPPPPPPPPP  PPP
UUU          UUU  EEEEEEEEEEEEEEEEE  TTTT TTTT TTTT TTTT  PPPPPPPPPPPPP  PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEEEEEEEEEEEEEE  TTT          PPPPPPPPPPPPP  PPP
UUU          UUU  EEEEEEEEEEEEEEE  TTT          PPPPPPPPPPPPP  PPP
UUU          UUU  EEEEEEEEEEEEEEE  TTT          PPPPPPPPPPPPP  PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUU          UUU  EEE          TTT          PPP          PPP
UUUUUUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEEEE  TTT          PPP
UUUUUUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEEEE  TTT          PPP
UUUUUUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEEEE  TTT          PPP

```

[illegible]


```
0000 1      .IDENT 'V04-000'
0000 74     $BEGIN RMSTEST3,009,__,RMSTEST,<RELATIVE TEST PROGRAM>,<GBL, LONG>
0000 75
0000 76 ;
0000 77
0000 78     .ENABL DBG
0000 79
0000 80 ::
0000 81 :: this program tests the relative file org
0000 82 ::
0000 83 ::
0000 84 ::
0000 85 ::
0000 86 :: macros:
0000 87 ::
0000 88
0000 89     .MACRO BUFF NAM,SIZE
0000 90 NAM'BUF::
0000 91     .BLKB SIZE
0000 92     NAM'BSZ==SIZE
0000 93     .ENDM BUFF
0000 94
0000 95 ::
0000 96 ::
0000 97 ::
0000 98
0000 99     .MACRO TYPE STRING, ?L
0000 100    STORE <STRING>
0000 101    BLBC VERBOSITY,L
0000 102    MOVL $$,TMPX,CMDORAB+RAB$RBF
0000 103    MOVW $$,TMPX1,CMDORAB+RAB$RSZ
0000 104    $PUT RAB=CMDORAB,ERR=REPORT_ERROR
0000 105    BSBW ERR
0000 106 L:
0000 107     .ENDM TYPE
0000 108
0000 109 ;
0000 110
0000 111     .MACRO WTTYPE STRING
0000 112     $WAIT CMDORAB
0000 113     TYPE <STRING>
0000 114     .ENDM WTTYPE
0000 115
0000 116 ;
0000 117
0000 118     .MACRO STORE STRING,PRE
0000 119     .SAVE
0000 120     .PSECT __,SRMSNAM
0000 121     $$,TMPX=,
0000 122     PRE ; store any carriage control info
0000 123     .ASCII %STRING%
0000 124     $$,TMPX1=.-$$,TMPX
0000 125     .RESTORE
0000 126     .ENDM STORE
```

```
0000 128
0000 129 ;
0000 130
0000 131 .MACRO FNM STRING
0000 132 STORE <STRING>
0000 133 MOVB $$$TMPX1,FAB$B_FNS+RELATIVE_FAB
0000 134 MOVL $$$TMPX,FAB$L_FNA+RELATIVE_FAB
0000 135 .ENDM
0000 136
0000 137 .MACRO BEGIN TSTNAM
0000 138 STORE <TSTNAM>
0000 139 MOVL $$$TMPX,BEG_DESCR+4 ; addr
0000 140 MOVL $$$TMPX1,BEG_DESCR ; len
0000 141 BSBW BEGPUT
0000 142 .ENDM BEGIN
0000 143 .MACRO FINISH TSTNAM
0000 144 STORE <TSTNAM>
0000 145 MOVL $$$TMPX,FIN_DESCR+4 ; addr
0000 146 MOVL $$$TMPX1,FIN_DESCR ; len
0000 147 BSBW FINPUT
0000 148 .ENDM FINISH
0000 149 .MACRO FIELD FLDNAM
0000 150 STORE <FLDNAM>
0000 151 MOVL $$$TMPX,FLD_DESCR+4 ; addr
0000 152 MOVL $$$TMPX1,FLD_DESCR ; len
0000 153 BSBW FLDPUT
0000 154 .ENDM FIELD
0000 155 .MACRO MBPT ?L
0000 156 BLBC VERBOSITY,L
0000 157 BPT
0000 158 L:
0000 159 .ENDM MBPT
0000 160
0000 161 ;
0000 162
```



```

00000000 164 .PSECT RMSTEST,GBL, LONG
0000 165 .ALIGN LONG
0000 166 T3START:
0000 167 RELATIVE FAB:
0000 168 $FAB FAC=<GET,PUT,DEL,UPD>,-
0000 169 DNM=<TST$DISK:.FIL;1>,-
0000 170 NAM=NAMBLK,-
0000 171 FOP=<DFW,SUP>,-
0000 172 ORG=REL,-
0000 173 RAT=CR,-
0000 174 MRS=52,-
0000 175 MRN=500,-
0000 176 ALQ=0,- ; alq=0 forces extend
0000 177 FSZ=4
0050 178
0050 179 RELATIVE_RAB:
0050 180 $RAB $FAB=RELATIVE_FAB,-
0050 181 UBF=RELBUF,-
0050 182 USZ=RELBSZ,-
0050 183 RBF=RELBUF,-
0050 184 RHB=HEAD,-
0050 185 MBF=3,-
0050 186 ROP=UIF
0094 187 ALLOC_XAB:
0094 188 $XABALL AID=0,-
0094 189 DEQ=4,-
0094 190 ALQ=4
00B4 191 COUNTER:
00B4 192 .BYTE 0
00B5 193 COUNT2: .BYTE 0
00B6 194 HEAD: .LONG 0
00BA 195 KEY: .LONG 0
00BE 196 RFATBL: .BLKQ 6
00EE 197 .ALIGN LONG
00F0 198 BUFF REL,200
029' 01B8 199 RFMSTR: .LONG RFML,RFMS
4F 01C0 200 RFMS: .ASCII 'OPENED RELATIVE FILE WITH FILE ORG !AD !/'
49 01CC
48 01D8
41 01E4
029 01E9 201 RFML=-RFMS

```

54	41	4C	45	52	20	44	45	4E	45	50	4F
54	49	57	20	45	4C	49	46	20	45	56	49
21	20	47	52	4F	20	45	4C	49	46	20	48
							2F	21	20	44	41

[illegible]

Address	Op Code	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

PSE

• R
\$AB
\$RM
\$

```
032D 253 REL_TEST:
032D 254
032D 255 ;
032D 256 ;routine to put relative thru its paces, and call the locking routine
032D 257 ;
032D 258
032D 259 $FAB_STORE FAB=RELATIVE_FAB,-
032D 260 ALQ=#0 ; make sure it's 0
0335 261 $CREATE FAB=RO,-
0335 262 ERR=REPORT_ERROR
FCB9' 30 0344 263 BSBW ERR
0347 264 $XABALL_STORE XAB=ALLOC_XAB,-
0347 265 ALQ=#4
0350 266 $FAB_STORE FAB=RELATIVE_FAB,-
0350 267 XAB=ALLOC_XAB
035B 268 $EXTEND FAB=RO,-
035B 269 ERR=REPORT_ERROR ; extend 4 blks, from alq
04 FC93' 30 036A 270 BSBW ERR
FC9B CF D1 036D 271 CMPL FAB$SL_STV+RELATIVE_FAB,#4 ; check returned stv
15 18 0372 272 BGEQ STVOK
04 FD17 CF D1 0374 273 FIELD <STV ( NOT = ALLOC QTY, AFTER EXTEND)>
15 18 0389 274 STVOK: CMPL XAB$SL_ALQ+ALLOC_XAB,#4 ; check alq
038E 275 BGEQ ALQOK
0390 276 FIELD <ALQ IN XAB ( NOT = ALLOC QTY, AFTER EXTEND)>
03A5 277 ALQOK: $CLOSE FAB=RELATIVE_FAB,-
03A5 278 ERR=REPORT_ERROR
FC47' 30 03B6 279 BSBW ERR
03B9 280 $XABALL_STORE XAB=ALLOC_XAB,-
03B9 281 ALQ=#0
03C1 282 $OPEN FAB=RELATIVE_FAB,-
03C1 283 ERR=REPORT_ERROR
05 FC2B' 30 03D2 284 BSBW ERR
FCCB CF D1 03D5 285 CMPL ALLOC_XAB+XAB$SL_ALQ,#5 ; alq=4 from extend + 1 since rel
15 18 03DA 286 BGEQ ALQOKT
03DC 287 FIELD <ALQ IN XAB ( NOT = ALLOC QTY, AFTER OPEN)>
01 FC2A CF 91 03F1 288 ALQOK1: CMPB RELATIVE_FAB+FAB$B_RFM,#FAB$C_FIX
08 13 03F6 289 BEQL REL_TEST2 ; on w/ it
03F8 290 $FAB_STORE FAB=RELATIVE_FAB,- ; if var clear xab
03F8 291 XAB=#0
```



```
0400 293 REL_TEST2:
0400 294
0400 295 ;
0400 296 ;entry point to bypass create
0400 297 ;
0400 298
0400 299 $CONNECT RAB=RELATIVE_RAB,-
0400 300 ERR=REPORT_ERROR
FBEC' 30 0411 301 BSBW ERR
0414 302
0414 303 ;
0414 304 ;do 26 sequential puts with 'bad' data into file
0414 305 ;then delete 5 of those records and do puts to put right data into them
0414 306 ;update the other records and then do gets to make sure it's all ok
0414 307 ;
0414 308
0414 309 MOVL #1,R9 ; r9 is record number
5B 59 01 DO 0417 310 MOVAL RELATIVE_RAB,R11 ; r11 is address of rab
FC35 CF DE 041C 311 TYPE <FILL FILE>
044B 312 PUT_RECORD_SEQ::
044B 313 MOVL #15,R6 ; all of length 15
FBCC 56 OF DO 044E 314 CMPB #FAB$C_FIX,FAB$B_RFM+RELATIVE_FAB
CF 01 91 0453 315 BNEQ 30$
03 12 0455 316 MOVL #52,R6 ; len of fixed rec. is 52
FC90 CF 56 31 56 34 DO 0458 317 30$: MOVC5 #0,(SP),#^A/1/,R6,RELBUF ; fill buffer for output
6E 00 2C 0460 318 ; put ascii '1's into records
22 AB 56 B0 0460 319 MOVW R6,RAB$W_RSZ(R11) ; give size of record
FC4D CF 59 DO 0464 320 MOVL R9,HEAD ; fill header in case it's vfc
0469 321
0469 322 ;
0469 323 ;seq mode is default
0469 324 ;
0469 325
0469 326 $PUT RAB=R11,-
0469 327 ERR=REPORT_ERROR
59 FB85' 30 0478 328 BSBW ERR
38 AB D1 047B 329 CMPL RAB$L_BKT(R11),R9 ; bkt should be rec. # on output
15 13 047F 330 BEQL BKT_OK
0481 331 FIELD <BKT IN RAB (RECORD NUMBER)>
B1 59 1A F3 0496 332 BKT_OK: AOBLEQ #26,R9,PUT_RECORD_SEQ ; keep going?
049A 333
049A 334 ;
049A 335 ;check one record just to be sure
049A 336 ;
049A 337
049A 338 MOVL #15,R9
30 AB 59 OF DO 049D 339 MOVAL KEY,RAB$L_KBF(R11)
1E AB 01 90 04A3 340 MOVB #RAB$C_KEY,RAB$B_RAC(R11)
FCOE CF 59 DO 04A7 341 MOVL R9,KEY
04AC 342 $GET RAB=R11,ERR=REPORT_ERROR
FB42' 30 04BB 343 BSBW ERR
01B7 30 04BE 344 BSBW CHK_BAD_DATA
04  AB 10 CA 04C1 345 BICL2 #RAB$M_OIF,RAB$L_ROP(R11) ; in case its set
59 05 DO 04C5 346 MOVL #5,R9
04C8 347 FIND_DEL:
04C8 348
04C8 349 ;
```



```

                                04C8 350 ;do some finds and deletes by keyed access
                                04C8 351 ;
                                04C8 352
FBED CF 59 D0 04C8 353          MOVL R9,KEY
                                04CD 354          $FIND RAB=R11,-
                                04CD 355          ERR=REPORT_ERROR
                                04DC 356          BSBW ERR
                                04DF 357          $DELETE RAB=R11,-
                                04DF 358          ERR=REPORT_ERROR
                                04EE 359          BSBW ERR
59 05 C0 04F1 360          ADDL #5,R9
1A 59 D1 04F4 361          CMPL R9,#26
CF 19 04F7 362          BLSS FIND_DEL

```

```
04F9 364
04F9 365 :
04F9 366 :done with deletes, now 'put' into the deleted records correct data
04F9 367 :
04F9 368 :
59 05 D0 04F9 369      MOVL    #5,R9
04FC 370 PUT_RECORD_KEY:
04FC 371
04FC 372 :
04FC 373 :try to get deleted records, hoping it fails
04FC 374 :then set nxr and get the deleted records and check them
04FC 375 :finally put the corrected (previously deleted)records
04FC 376 :
04FC 377 :
00C8 8F 00 6E 00 2C 04FC 378      MOVC5    #0,(SP),#0,#200,RELBUF ; clr relbuf, to make sure
      FBEA CF 59 D0 0503 379
      FBAF CF 59 D0 0506 379      MOVL    R9,KEY ; gets by key
00000000'8F 50 D1 050B 380      $GET    RAB=R11 ; hope it fails
      1E 13 0514 381      CMPL    R0,#RMS$_RNF ; record not found?
      5A 5B D0 051B 382      BEQL    ERR_OK
      FAC8' 30 051D 383      FIELD    <RETURNED ERROR CODE>
      FAC5' 30 0532 384      MOVL    R11,R10
00800000 8F C8 0535 385      BSBW    REPORT_ERR
      FB10 CF 0538 386      BSBW    ERR
      053B 387 ERR_OK: BISL    #RAB$M_NXR,- ; get non-existent record
      0541 388      RAB$M_NXR,-
      0544 389      $GET    RAB$M_NXR,- ; this should work
      0544 390      ERR=REPORT_ERROR
      FAAA' 30 0553 391      BSBW    ERR
      011F 30 0556 392      BSBW    CHK_BAD_DATA ; check it out
00800000 8F CA 0559 393      BICL    #RAB$M_NXR,- ; clear bit
      FAF2 CF 055F 394      RAB$M_NXR,-
      0222 30 0562 395      BSBW    RAB$M_NXR,-
FB50 CF 59 D0 0565 396      MOVL    R9,KEY ; fills buffer,rsz,head
      056A 397      $PUT    RAB=R11,- ; keyed access
      056A 398      ERR=REPORT_ERROR
      FA84' 30 0579 399      BSBW    ERR
      59 05 C0 057C 400      ADDL    #5,R9
      1A 59 D1 057F 401      CMPL    R9,#26
      03 18 0582 402      BGEQ    10$
      FF75 31 0584 403      BRW     PUT_RECORD_KEY
      0587 404 10$:
      0587 405
      0587 406 :
      0587 407 :all done with that
      0587 408 :
      0587 409
59 01 D0 0587 410      MOVL    #1,R9
      058A 411
      058A 412 :
      058A 413 :do updates on all other records, by keyed access
      058A 414 :
      058A 415 :
      058A 416 UPDATE_RECORD:
      058A 417      CLRL    R10
      56 52 59 05 7B 058C 418      EDIV    #5,R9,R2,R6
      56 D5 0591 419      TSTL    R6 ; is it 5,10,15,20,25
```

```

F1 59 07 12 0593 420 BNEQ 20$
1A F3 0595 421 AOBLEQ #26,R9,UPDATE_RECORD ; if so, skip it
0030 31 0599 422 BRW NO_MORE ; all done
FB19 CF 59 D0 059C 423 20$: MOVL R9,KEY
05A1 424 $FIND RAB=R11,-
05A1 425 ERR=REPORT_ERROR
FA4D' 30 05B0 426 BSBW ERR
01D1 30 05B3 427 BSBW SETUP ; set up for put
05B6 428 $UPDATE RAB=R11,-
05B6 429 ERR=REPORT_ERROR
FA38' 30 05C5 430 BSBW ERR
1A F3 05C8 431 AOBLEQ #26,R9,UPDATE_RECORD
BE 59 00 90 05CC 432 NO_MORE: MOVB #RAB$C_SEQ,RAB$B_RAC+RELATIVE_RAB
FA9D CF 00 90 05CC 433 TYPE <VERIFY CONTENTS OF FILE>
05D1 434 $REWIND RAB=R11,- ; can now do gets
0600 435 ERR=REPORT_ERROR
0600 436 BSBW ERR
F9EE' 30 060F 437 BSBW DO_SOME_GETS
00AA 30 0612 438 $DISCONNECT RAB=R11,-
0615 439 ERR=REPORT_ERROR
F9D9' 30 0624 440 BSBW ERR
0627 441 $CONNECT RAB=R11,-
0627 442 ERR=REPORT_ERROR
F9C7' 30 0636 443 BSBW ERR
0083 30 0639 444 BSBW DO_SOME_GETS
063C 445 $DISCONNECT RAB=R11,-
063C 446 ERR=REPORT_ERROR
F9B2' 30 064B 447 BSBW ERR
064E 448 $CLOSE FAB=RELATIVE_FAB,-
064E 449 ERR=REPORT_ERROR
F99E' 30 065F 450 BSBW ERR
0662 451 ; tell locking test which file
F9CE CF 90 0662 452 MOVB FAB$B_FNS+RELATIVE_FAB,-
00000034'EF 0666 453 FAB$B_FNS+LOCK_FAB
F9BD CF D0 066B 454 MOVL FAB$L_FNA+RELATIVE_FAB,-
0000002C'EF 066F 455 FAB$L_FNA+LOCK_FAB
F989' 30 0674 456 BSBW RMT$TEST_5A ; do locking tests
05 0677 457 RSB
067B 458
067B 459
067B 460 ;subroutine to check 'bad' data ( 1st pass of puts)
067B 461 ;r11 is pointer to relative rab
067B 462 ;routine checks rsz and contents of record, now in buffer
067B 463
067B 464
067B 465
067B 466
067B 467 CHK_BAD_DATA:
067B 468 MOVL #15,R6 ; len of non-fixed records
067B 469 CMPB #FAB$C_FIX,-
067D 470 RELATIVE_FAB+FAB$B_RFM
0680 471 BNEQ 10$ ; if fix len is 52
0682 472 MOVL #52,R6
0685 473 10$: CMPW R6,RAB$W_RSZ(R11) ; check rsz
0689 474 BEQL RSZ_OK
068B 475 FIELD <RSZ>
06A0 476 RSZ_OK:
```


RMSTEST3
009

RELATIVE TEST PROGRAM ;

F 8

16-SEP-1984 01:47:03 VAX/VMS Macro V04-00
5-SEP-1984 04:21:48 [UETP.SRC]RMSTEST3.MAR;1

Page 10
(9)

6E	00	31	28	BB	56	2D	06A0	477	CMPCS	R6, @RABSL_RBF(R11), #A/1/, #0, (SP)
					15	13	06A7	478	BEQL	REC_OK
							06A9	479	FIELD	<RECORD>
							06BE	480	REC_OK:	
					05		06BE	481	RSB	

54	F9FB CF	DE	06BF	483	DO_SOME_GETS::		
	F9EC CF	94	06BF	484	MOVAL	RFATBL,R4	: r4 is index to rfatbl
	F9E9 CF	94	06C4	485	CLRB	COUNTER	: record number
	57 1A	DO	06C8	486	CLRB	COUNT2	
5B	F97D CF	DE	06CC	487	MOVL	#26,R7	: 1st pass-r7 is # of rec.
1E	AB 00	90	06CF	488	MOVAL	RELATIVE_RAB,R11	: pointer to rab
	00E1	30	06D4	489	MOVB	#RAB\$C_SEQ,RAB\$B_RAC(R11)	: do all sequential gets
1E	AB 02	90	06D8	490	BSBW	GET_RECORD_SEQ	
54	F9DB CF	DE	06DB	491	MOVB	#RAB\$C_RFA,RAB\$B_RAC(R11)	: do some gets by rfa
	0180	30	06DF	492	MOVAL	RFATBL,R4	
1E	AB 01	90	06E4	493	BSBW	GET_RECORD_RFA	
	01A2	30	06E7	494	MOVB	#RAB\$C_KEY,RAB\$B_RAC(R11)	: do some gets by key
1E	AB 00	90	06EB	495	BSBW	GET_RECORD_KEY	
	57 0D	DO	06EE	496	MOVB	#RAB\$C_SEQ,RAB\$B_RAC(R11)	: do some seg gets
54	F9DD CF	DE	06F2	497	MOVL	#13,R7	: 2nd pass-r7 is # of rec.
F9B5	CF 0D	90	06F5	498	MOVAL	RFATBL+24,R4	: 24=8.*3, starting in the middle
	F9B2 CF	94	06FA	499	MOVB	#13,COUNTER	: ditto
	00B6	30	06FF	500	CLRB	COUNT2	
		05	0703	501	BSBW	GET_RECORD_SEQ	
			0706	502	RSB		

```
0707 504
0707 505
0707 506 : subroutines to do gets and checks
0707 507 :
0707 508
0707 509 CHECK_REC:
0707 510
0707 511 :
0707 512 : r9 is the record number
0707 513 : as before r11 is the addr of the rab
0707 514 :
0707 515
0707 516 CMPB #FAB$C_VFC,FAB$B_RFM+RELATIVE_FAB
0707 517 BNEQ NO_HEADER
0707 518 100$: CMPL R9,@RAB$L_RHB(R11) : compare header
0707 519 BEQL NO_HEADER
0707 520 FIELD <HEADER OF A VFC RECORD>
0707 521 NO_HEADER:
0707 522 CMPL R9,#26 : is it last record?
0707 523 BNEQ 10$
0707 524 MOVL #26,R6 : length is 26
0707 525 BRB 20$
0707 526 10$: CLRL R10 : quadword divide
0707 527 EDIV #26,R9,R2,R6 : r6 = r9 mod 26, rec. size
0707 528 20$: ADDL3 #^A/A/-1,R6,R5 : r5 is char.
0707 529 CMPB #FAB$C_FIX,FAB$B_RFM+RELATIVE_FAB
0707 530 BNEQ GOT_RS
0707 531 MOVL #52,R6 : mrs for fixed is 26
0707 532 GOT_RS:
0707 533 CMPW R6,RAB$W_RSZ(R11) : check rsz
0707 534 BEQL OK_RSZ
0707 535 FIELD <RSZ FIELD IN RAB>
0707 536 OK_RSZ:
0707 537 10$: CMPC5 R6,@RAB$L_RBF(R11),R5,#0,(SP)
0707 538 BNEQ BADREC
0707 539 RSB : return
0707 540 BADREC:
0707 541 FIELD <RECORD>
0707 542 RSB

F913 CF 03 91 0707 516
18 12 0707 517
2C BB 59 D1 0707 518 100$:
15 13 0707 519 BEQL
0707 520 FIELD
1A 59 D1 0707 521 NO_HEADER:
05 12 0707 522 CMPL
56 1A D0 0707 523 BNEQ
07 11 0707 524 MOVL
5A D4 0707 525 BRB
55 56 52 59 1A 7B 0707 526 10$:
00000040 8F C1 0707 527 EDIV
F8D8 CF 01 91 0707 528 20$:
03 12 0707 529 CMPB
56 34 D0 0707 530 BNEQ
22 AB 56 B1 0707 531 MOVL
15 13 0707 532 GOT_RS:
0707 533 CMPW
0707 534 BEQL
0707 535 FIELD
6E 00 55 28 BB 56 2D 0707 536 OK_RSZ:
01 12 0707 537 10$:
05 05 0707 538 BNEQ
0707 539 RSB
0707 540 BADREC:
0707 541 FIELD
05 0786 542 RSB
```



```
0787 544 SETUP:
0787 545
0787 546 :
0787 547 : routine to do setup for puts of correct data
0787 548 : r9 is record number -- input
0787 549 : output -- relbuf is filled in with correct char
0787 550 : -- head is filled in, and rsz is also
0787 551 :
0787 552
1A 59 D1 0787 553 CMPL R9,#26
05 12 078A 554 BNEQ 10%
56 1A D0 078C 555 MOVL #26,R6 ; len of last rec. is 26
07 11 078F 556 BRB 20%
5A D4 0791 557 10%: CLRL R10
55 56 52 59 1A 7B 0793 558 EDIV #26,R9,R2,R6 ; r6 is rec. # mod 26
55 56 00000040 8F C1 0798 559 20%: ADDL3 #^A/A/-1,R6,R5 ; r5 is char. to fill buffer
F87A CF 01 91 07A0 560 CMPB #FAB$C_FIX,-
03 12 07A2 561 FAB$B_RFM+RELATIVE_FAB
56 34 D0 07A5 562 BNEQ 30%
F93E CF 56 55 6E 00 2C 07AA 564 30%: MOVL #52,R6 ; len. of fixed rec. is 52
22 AB 56 80 07B2 565 MOVCS #0,(SP),R5,R6,RELBUF ; fill relbuf
F8FB CF 59 D0 07B6 566 MOVW R6,RAB$W_RSZ(R11) ; fill rsz
05 07BB 567 MOVL R9,HEAD ; fill header, in case vfc
RSB
```

```
00000000'BF 50 D1 07BC 569
                    0F 12 07BC 570 GET_RECORD_SEQ:
57 F8E3 CF 91 07BC 571 :
                    01 12 07BC 572 :
                    05 05 07BC 573 :
                    07D6 574 :
                    07D6 575 :
                    07D9 576 $GET R11
                    07DC 577 CMPL R0,#RMS$_EOF
                    07DD 578 BNEQ MORE
                    07DE 579 CMPB COUNT2,R7
                    07DF 580 BNEQ BADNR
                    07E0 581 RSB
BADNR: 07E1 582
07E2 583 MOVL R11,R10
07E3 584 BSBW EOFPUT
07E4 585 RSB
07E5 586 MORE:
07E6 587 BLBS R0,10$
07E7 588 MOVL R11,R10
07E8 589 BSBW REPORT_ERR
07E9 590 BSBW ERR
57 F8C8 CF 91 07EA 591 10$: CMPB COUNT2,R7
07EB 592 BLEQ 20$
07EC 593 BRB BADNR
07ED 594 20$: INCB COUNTER
07EE 595 INCB COUNT2
59 F8BE CF 96 07EF 596 MOVZBL COUNTER,R9
F8B8 CF 96 07F0 597 CMPL R9,RAB$_BKT(R11)
59 F8B6 CF 9A 07F1 598 BEQL RNOK
38 AB 59 D1 07F2 599 FIELD <BKT FIELD IN RAB>
15 13 0803 600 RNOK:
FEEA 30 0804 601 BSBW CHECK_REC
52 50 59 SA D4 0805 602 CLRL R10 ; quad word divide
01 52 78 0806 603 EDIV #5,R9,R0,R2
03 13 D1 0807 604 CMPL R2,#1
FF90 31 0808 605 BEQL SAV_RFA
0809 606 BRW GET_RECORD_SEQ ; continue
080A 607 SAV_RFA:
080B 608
080C 609 :
080D 610 : save record numbers 1,6,11,16,21,26 on 1st pass
080E 611 : check record numbers 16,21,26 on 2nd pass
080F 612 :
0810 613
0811 614 BLBC R7,SAV ; which pass?
64 10 AB 2A 57 E9 0812 615 CMPC3 #6,RAB$_RFA(R11),(R4) ; 2nd, check them
06 29 0813 616 BEQL RFA_OK
1D 13 0814 617 FIELD <RFA>
0815 618 MBPT
0816 619 RFA_OK:
0817 620 ADDL2 #8,R4
54 08 00 0818 621 BRW GET_RECORD_SEQ ; on to next record
FF63 31 0819 622 SAV:
0820 623 MOVQ RAB$_RFA(R11),(R4)+ ; 1st pass, save them
84 10 AB 7D 0821 624 SUBL3 #1,R4-R2
52 54 01 C3 0822 625 MOVB R9,(R2) ; also store record number
62 59 90 0823
```

```
FF55 31 0864 626 BRW GET_RECORD_SEQ
0867 627
0867 628 GET_RECORD_RFA:
0867 629
0867 630 :
0867 631 : get records by rfa
0867 632 :
0867 633
000000EE'8F 54 D1 0867 634 CMPL R4,#RFATBL+48 : more rfa entries
1F 18 086E 635 BGEQ END_OF_RFA
10 AB 84 7D 0870 636 MOVQ (R4)+,RAB$W_RFA(R11) : load rab w/ rfa
0874 637 $GET RAB=R11,ERR=REPORT_ERROR
59 F77A' 30 0883 638 BSBW ERR
17 AB 90 0886 639 MOVB RAB$W_RFA+7(R11),R9 : get record number, as stored
FE7A 30 088A 640 BSBW CHECK_REC : r9 is now rec. #
DB 11 088D 641 BRB GET_RECORD_RFA
05 088F 642 END_OF_RFA:
088F 643 RSB
0890 644
0890 645 GET_RECORD_KEY:
0890 646
0890 647 :
0890 648 : get records by key
0890 649 :
0890 650
30 AB F826 CF DE 0890 651 MOVAL KEY,RAB$L_KBF(R11)
F81F CF 01 D0 0896 652 MOVL #1,KEY : get 1st record
0010 30 089B 653 BSBW GETCHK : get and check
F817 CF 1A D0 089E 654 MOVL #26,KEY
0008 30 08A3 655 BSBW GETCHK
F80F CF 0D D0 08A6 656 MOVL #13,KEY
0000 31 08AB 657 BRW GETCHK
08AE 658
08AE 659 GETCHK:
08AE 660 $GET RAB=R11,-
08AE 661 ERR=REPORT_ERROR
59 F740' 30 08BD 662 BSBW ERR
F7F6 CF D0 08C0 663 MOVL KEY,R9
FE3F 31 08C5 664 BRW CHECK_REC
08C8 665 .END
```


RMSTEST3
Symbol table

RELATIVE TEST PROGRAM ;

L 8

16-SEP-1984 01:47:03 VAX/VMS Macro V04-00
5-SEP-1984 04:21:48 [UETP.SRC]RMSTEST3.MAR;1

Page 16
(15)

\$\$PSECT_EP
\$\$TAB
\$\$TABEND
\$\$TMP
\$\$TMP1
\$\$TMP2
\$\$TMPX
\$\$TMPX1
\$\$RMSTEST
\$\$RMS_PBUGCHK
\$\$RMS_TBUGCHK
\$\$RMS_UMODE
..AFLG
..FLG
..MOD
..TYP
..LEN
ALLOC_XAB
ALQOK
ALQOK1
BADNR
BADREC
BEGPUT
BEG_DESCR
BKT_OK
CHECK_REC
CHK_BAD_DATA
CMDORAB
COUNT2
COUNTER
DO SOME GETS
END_OF_RFA
EOFPUT
ERR
ERR OK
FAB\$B_DNS
FAB\$B_FNS
FAB\$B_FSZ
FAB\$B_RFM
FAB\$C_BID
FAB\$C_BLN
FAB\$C_FIX
FAB\$C_REL
FAB\$C_VAR
FAB\$C_VFC
FAB\$SL_ALQ
FAB\$SL_DNA
FAB\$SL_FNA
FAB\$SL_FOP
FAB\$SL_STV
FAB\$SL_XAB
FAB\$M_CIF
FAB\$M_SUP
FAB\$V_CHAN_MODE
FAB\$V_CR
FAB\$V_DEL
FAB\$V_DFW

= 00000000
= 00000094 R D 01
= 00000084 R D 01
= 00000002
= 00000002
= 0000005B
= 00000191 R D 04
= 00000003
= 0000001E
= 00000010
= 00000008
= 00000004
= 00000000 D
= 00000001 D
= 00000001 D
= 000000CF
= 00000004
00000094 R D 01
000003A5 R D 01
000003F1 R D 01
000007D6 R D 01
00000771 R D 01
***** X 01
***** X 01
00000496 R D 01
00000707 R D 01
00000678 R D 01
***** X 01
000000B5 R D 01
00000084 R D 01
000006BF RG D 01
0000088F R D 01
***** X 01
***** X 01
0000053B R D 01
= 00000035 D
= 00000034 D
= 0000003F D
= 0000001F D
= 00000003 D
= 00000050 D
= 00000001 D
= 00000010 D
= 00000002 D
= 00000003 D
= 00000010 D
= 00000030 D
= 0000002C D
= 00000004 D
= 0000000C D
= 00000024 D
= 02000000 D
= 00000004 D
= 00000002 D
= 00000001 D
= 00000002 D
= 00000005 D

FAB\$V_FILE_MODE
FAB\$V_GET
FAB\$V_LNM_MODE
FAB\$V_PUT
FAB\$V_SUP
FAB\$V_UPD
FAB\$W_GBC
FIND_DEL
FINPUT
FIN_DESCR
FLDPUT
FLD_DESCR
GETCHK
GET_RECORD_KEY
GET_RECORD_RFA
GET_RECORD_SEQ
GOT_RS
HEAD
KEY
LOCK_FAB
MORE
NAMBLK
NO_HEADER
NO_MORE
OK_RSZ
PUT_RECORD_KEY
PUT_RECORD_SEQ
RAB\$B_RAC
RAB\$C_BID
RAB\$C_BLN
RAB\$C_KEY
RAB\$C_RFA
RAB\$C_SEQ
RAB\$SL_BKT
RAB\$SL_CTX
RAB\$SL_KBF
RAB\$SL_RBF
RAB\$SL_RHB
RAB\$SL_ROP
RAB\$M_NXR
RAB\$M_UIF
RAB\$V_UIF
RAB\$W_RFA
RAB\$W_RSZ
REC_OR
RELATIVE_FAB
RELATIVE_RAB
RELBSZ
RELBUF
REL_TEST
REL_TEST2
REPORT_ERR
REPORT_ERROR
RFATBL
RFA_OK
RFMC
RFMS

= 00000004 D
= 00000001 D
= 00000000 D
= 00000000 D
= 00000002 D
= 00000003 D
= 00000048 D
000004C8 R D 01
***** X 01
***** X 01
***** X 01
***** X 01
000008AE R D 01
00000890 R R D 01
00000867 R R D 01
000007BC R R D 01
0000074C R R D 01
000000B6 R R D 01
000000BA R D 01
***** X 01
000007DD R D 01
***** X 01
00000729 R D 01
000005CC R R D 01
00000767 R R D 01
000004FC R D 01
0000044B RG D 01
= 0000001E D
= 00000001 D
= 00000044 D
= 00000001 D
= 00000002 D
= 00000000 D
= 00000038 D
= 00000018 D
= 00000030 D
= 00000028 D
= 0000002C D
= 00000004 D
= 00800000 D
= 00000010 D
= 00000004 D
= 00000010 D
= 00000022 D
000006BE R D 01
00000000 R R D 01
00000050 R D 01
= 000000C8 G D 01
000000F0 RG D 01
0000032D R D 01
00000400 R D 01
***** X 01
***** X 01
000000BE R D 01
00000853 R D 01
= 00000029 D
000001C0 R D 01

RMSTEST3
Symbol table

RELATIVE TEST PROGRAM :

M 8

16-SEP-1984 01:47:03 VAX/VMS Macro V04-00
5-SEP-1984 04:21:48 [UETP.SRC]RMSTEST3.MAR;1

Page 17
(15)

RFMSTR	000001B8	R	D	01
RMS\$_CREATED	*****	X		01
RMS\$_EOF	*****	X		01
RMS\$_RNF	*****	X		01
RMT\$TEST_3A	000001E9	RG	D	01
RMT\$TEST_5A	*****	X		01
RNOK	0000081A	R	D	01
RSZ_OK	000006A0	R	D	01
SAV	00000859	R	D	01
SAV_RFA	0000082C	R	D	01
SETOP	00000787	R	D	01
STVOK	00000389	R	D	01
SYSS\$CLOSE	*****	GX		01
SYSS\$CONNECT	*****	GX		01
SYSS\$CREATE	*****	GX		01
SYSS\$DELETE	*****	GX		01
SYSS\$DISCONNECT	*****	GX		01
SYSS\$EXTEND	*****	GX		01
SYSS\$FIND	*****	GX		01
SYSS\$GET	*****	GX		01
SYSS\$OPEN	*****	GX		01
SYSS\$PUT	*****	GX		01
SYSS\$REWIND	*****	GX		01
SYSS\$UPDATE	*****	GX		01
T3START	00000000	RG	D	01
UPDATE_RECORD	0000058A	R	D	01
VERBOSITY	*****	X		01
XABS\$B_AID	= 00000017		D	
XABS\$B_AOP	= 00000008		D	
XABS\$B_BKZ	= 00000016		D	
XABS\$C_ALL	= 00000014		D	
XABS\$C_ALLLEN	= 00000020		D	
XABS\$L_ALQ	= 00000010		D	
XABS\$L_LOC	= 0000000C		D	
XABS\$L_NXT	= 00000004		D	
XABS\$W_DEQ	= 00000014		D	
XABS\$W_RF10	= 00000018		D	
XABS\$W_RF12	= 0000001A		D	
XABS\$W_RF14	= 0000001C		D	
XABS\$W_VOL	= 0000000A		D	

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes													
. ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE			
. RMSTEST	000008C8 (2248.)	01 (1.)	NOPIC	USR	CON	REL	GBL	NOSHR	EXE	RD	WRT	NOVEC	LONG			
\$ABS\$	00000000 (0.)	02 (2.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE			
\$RMSNAM	0000000F (15.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE			
__RMSNAM	00000194 (404.)	04 (4.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE			

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	38	00:00:00.09	00:00:00.38
Command processing	133	00:00:00.60	00:00:03.73
Pass 1	299	00:00:11.36	00:00:25.25
Symbol table sort	0	00:00:00.55	00:00:01.11
Pass 2	118	00:00:02.85	00:00:06.66
Symbol table output	19	00:00:00.12	00:00:00.16
Psect synopsis output	2	00:00:00.04	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	611	00:00:15.61	00:00:37.33

The working set limit was 1500 pages.
54514 bytes (107 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 441 non-local and 20 local symbols.
665 source lines were read in Pass 1, producing 46 object records in Pass 2.
64 pages of virtual memory were used to define 49 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	35
TOTALS (all libraries)	35

870 GETS were required to define 35 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RMSTEST3/OBJ=OBJ\$:RMSTEST3 MSRC\$:RMSTEST3/UPDATE=(ENH\$:RMSTEST3)+EXECML\$/LIB

0409

**DIGITAL
CONFIDE**

EQUIPMENT
NTIAL AND

CORPORATION
PROPRIETARY